

Ottawa Hull K1A 0C9

(21) (A1)	2,160,087
(22)	1995/10/06
(43)	1996/04/11

(51) Int.Cl. ⁶ H04N 7/50

(19) (CA) APPLICATION FOR CANADIAN PATENT (12)

(54) Process for Coding or Decoding Motion Vectors and Coding or Decoding Device Implementing the Said Process

(72) Kerdranvat, Michel - France ;

(71) Thomson multimedia S.A. - France ;

(30) (FR) 94 12083 1994/10/10

(57) 16 Claims

Notice: This application is as filed and may therefore contain an incomplete specification.



Industrie Canada Industry Canada

OPIC - CIPO 191

Canada

2160087

ABSTRACT

**PROCESS FOR CODING OR DECODING MOTION VECTORS AND CODING OR
DECODING DEVICE IMPLEMENTING THE SAID PROCESS**

The invention relates to a process for the differential coding of a motion vector (MV0) associated with a macro-block (M0). This process comprises the steps:

- of determining the components (MV0x, MV0y) of the said motion vector (MV0),
- of determining a prediction vector, each component (Px, Py) of which is equal to the median value of the corresponding components of at least three candidate motion vectors (MV1, MV2, MV3), a candidate motion vector being a vector associated with a previously coded macro-block,
- of subtracting the components thus obtained from the components of the vector to be coded.

The invention also relates to a process for decoding motion vectors coded in the manner described above, as well as to the implementation devices.

The invention applies to the field of image compression.

Figure 3

2160087

- 1 -

**PROCESS FOR CODING OR DECODING MOTION VECTORS AND CODING OR
DECODING DEVICE IMPLEMENTING THE SAID PROCESS**

The present invention relates to a process for coding motion vectors in an image compression system. The invention applies especially, but not uniquely, to the coding of video images in respect of telecommunication channels rated at less than 64 kbit/s.

The invention also relates to a decoding process, to a coding device and to a decoding device.

An image compression device, or coder, is described for example in Recommendation H.261 of the Comité Consultatif International Télégraphique et Téléphonique (CCITT), especially with reference to Figure 3 ("source coder") of this document.

The device is based on the processing of the image in pixel blocks or macro-blocks. A block is composed of a matrix of eight times eight pixels, whereas a macroblock comprises four blocks. A macro-block can be coded in one of two modes termed "inter" or "intra", all the blocks making up the macro-block being coded in the same mode. In the case of "intra" coding, the blocks are subjected to a discrete cosine transformation, the coefficients thus obtained being quantized, and then coded in accordance with a variable-length code process ("VLC" coding). In the case of "inter" coding, it is the difference between the blocks and blocks of like position in a reference image which is subjected to the discrete cosine transformation.

To improve the performance of the coding in the "inter"

- 2 -

mode, motion compensation processes are employed. A process of this type makes it possible to determine, in the reference image or a window thereof, a macro-block which will allow the effective "inter" coding of the macro-block to be coded of the current image. Various selection criteria exist for determining such a macro-block in the reference image. The relative position of the reference macro-block with respect to the position of the macro-block to be coded in the current image is determined by what is called a motion vector. This motion vector is transmitted in the header of the coded macro-block.

A decoder also possesses a memory containing the reference image. Knowing the motion vector and the coefficients corresponding to the coded macro-block, it will be able to restore the values of the pixels of the original macro-block. These values may contain errors, especially a quantization error.

The motion vectors are coded differentially with respect to a prediction vector. This vector is equal to the motion vector of the macro-block lying immediately to the left of the macro-block to be coded or decoded. The horizontal H and vertical V components of a vector are coded separately. According to a known process, the prediction vector is reinitialized to (0,0) in the following two cases:

- if the macro-block immediately to the left of the macro-block to be coded or decoded was coded in "intra" mode,
- if the macro-block to be coded lies on the left edge of the image.

The subject of the invention is a process for the

- 3 -

differential coding of a motion vector associated with a macro-block, characterized in that it comprises the steps:

- of determining the components of the said motion vector,
- of determining a prediction vector, each component of which is equal to the median value of the corresponding components of at least three candidate motion vectors, a candidate motion vector being a vector associated with a previously coded macro-block,
- of subtracting the components thus obtained from the components of the vector to be coded.

According to a particular embodiment, the components of a candidate motion vector are regarded as being zero when determining the prediction vector if the macro-block associated with this candidate motion vector lies outside the image.

According to a particular embodiment, the components of a candidate motion vector are regarded as being zero when determining the prediction vector if the macro-block associated with this candidate motion vector was coded in "intra" mode.

According to a particular embodiment, the macro-blocks associated with the candidate motion vectors are macro-blocks adjacent to the macro-block of which a vector is to be coded.

According to a particular embodiment, the macro-blocks associated with the candidate motion vectors form part of the macro-blocks lying above or to the left of the macro-block of which a vector is to be coded.

According to a particular embodiment, the macro-blocks associated with the candidate motion vectors comprise the macro-

2160087

- 4 -

block lying immediately to the left, the macro-block lying immediately above and the macro-block lying above and to the right of the macro-block of which a vector is to be coded.

The subject of the invention is also a process for the decoding of a motion vector coded by differential coding, characterized in that it comprises the steps:

- of extracting the differences in components corresponding to this vector,

- of determining a prediction vector, each component of which is equal to the median value of the corresponding components of vectors associated with at least three previously decoded macro-blocks,

- of adding the said differences in components to the components of the prediction vector.

According to a particular embodiment, the components of a candidate motion vector are regarded as being zero when determining the prediction vector if the macro-block associated with this candidate motion vector lies outside the image.

According to a particular embodiment, the components of a candidate motion vector are regarded as being zero when determining the prediction vector if the macro-block associated with this candidate motion vector was coded in "intra" mode.

According to a particular embodiment, the macro-blocks associated with the candidate motion vectors are macro-blocks adjacent to the macro-block of which a vector is to be decoded.

According to a particular embodiment, the macro-blocks associated with the candidate motion vectors form part of the

2160087

- 5 -

macro-blocks lying above or to the left of the macro-block of which a vector is to be decoded.

According to a particular embodiment, the macro-blocks associated with the candidate motion vectors comprise the macro-block lying immediately to the left, the macro-block lying immediately above and the macro-block lying above and to the right of the macro-block of which a vector is to be decoded.

The subject of the invention is also an image compression device including a motion estimation processor producing motion vectors, characterized in that, with the said vectors being coded with respect to a prediction vector, the said device comprises:

- means for determining a prediction vector, each component of which is equal to the median value of the corresponding components of at least three candidate motion vectors, a candidate motion vector being a vector associated with a previously coded macro-block,

- means for subtracting the components thus obtained from the components of the vector to be coded,

- means for coding the values obtained through the said subtraction.

The subject of the invention is also a device for decompressing images compressed with the aid of a motion compensation process, characterized in that it comprises:

- a buffer register receiving compressed data,
- demultiplexing means connected to the said register,
- dequantization means connected to the demultiplexing means,

2160087

- 6 -

- means of inverse transformation of demultiplexed and dequantized coefficients,

- means for extracting data relating to the motion vectors coded according to a differential coding with respect to a prediction vector,

- means for determining the prediction vector on the basis of the motion vectors corresponding to previously decoded macro-blocks.

According to a particular embodiment, the compression, respectively decompression device implements the coding, respectively decoding process.

The invention will be better understood through the description of a non-limiting preferred particular example illustrated by the figures among which:

- Figure 1 is a functional diagram of an image coder implementing the invention,

- Figure 2 represents the numbering of the macro-blocks of an image with the CIQF format,

- Figure 3 illustrates the processing of candidate vectors corresponding to macro-blocks lying beyond the boundaries of the image,

Figure 4 is a functional diagram of a decoder implementing the process according to the invention.

A video image coder is represented in Figure 1. This coder comprises a video input 1 feeding a motion compensation processor 10 (associated with a reference memory 2), a subtractor 3 and one of the inputs of a multiplexer 4 with two inputs. The second input

2160087

- 7 -

of the multiplexer 4 is connected to the output of the subtractor 3. The output of the multiplexer 4 is linked to a discrete cosine transformation processor (DCT processor 5).

In "intra" mode, the subtractor is not used. The values of pixels are transmitted directly from the input 1 to the DCT processor 5.

In "inter" mode, it is the difference between the signal at the input of the device and a portion of the reference image stored in the memory 2 which is transmitted to the DCT processor 5.

The output of the DCT processor 5 is connected to a quantization circuit 6, the output of which is connected on the one hand to a dequantization circuit 7 followed by an inverse DCT processor 8, and on the other hand to a coding and multiplexing circuit 14. The coding circuit carries out, in particular, the VLC coding of the motion vector differences, the VLC coding of the DCT transform coefficients and the assembling of the various data in accordance with the structure required.

A block to be coded is subjected to the DCT transformation, whether it be an "inter" or "intra" block, that is to say one containing pixel values or differences in pixel values respectively. The DCT coefficients obtained are quantized and then coded in accordance with VLC coding. Such VLC codings, as well as the associated processes (block scan direction, coding of coefficient pairs and of lengths of strings of zero coefficients) will not be described in further detail since they are well known to those skilled in the art. The coded coefficients are inserted

- 8 -

into the video multiplex in an appropriate manner.

The quantized coefficients are subjected to dequantization and then to inverse DCT transformation. This is the process to which these same data will be subjected in the decoder, the resulting pixel values containing the same errors, at least in part, notably the quantization errors. In the coder, these decoded data serve to update the reference image stored in the memory 2.

The output of the memory 2 is connected to an input of the subtractor 3, possibly across a spatial loop filter 9. The output of the inverse DCT processor 8 is connected to the reference memory 2 across a two-input adder 11. The second input of the adder 11 is connected to the output of a second multiplexer 12 with two inputs, one of which is connected to the output of the loop filter 9, and the other of which is not connected up, delivering zero pixel values. In "intra" mode, the output of the multiplexer 12 delivers the zero values. The two multiplexers 4 and 12 are controlled simultaneously by a control unit 13 which also handles the quantizing of the coefficients, notably as a function of the degree of fill of the video buffer register (not illustrated).

The motion compensation processor compares macro-blocks originating from the input 1 with macro-blocks of the reference image, and determines on the one hand whether a macro-block is to be coded in intra or inter mode. Such a decision can be made by evaluating the value of an error function as the sum of the squares of the pixel differences in the case of the "inter" mode with respect to threshold values.

2160087

- 9 -

The coding in "intra" mode is enforced from time to time if it is not chosen for a given number of images for one and the same macro-block. This is done in order to limit the accumulation of reconstruction errors within the inverse DCT transformation. For example, following recommendation H.261 mentioned earlier, such enforcement is carried out at least every 132 transmissions of one and the same block.

Motion compensation or estimation is employed in particular at the level of the processor 10. There are numerous processes for searching for the best motion vector and as many variants thereof: exhaustive comparison of blocks, hierarchical comparison processes etc. These processes can be implemented in particular by microprocessors, special-purposes processors etc.

According to the present example, the structure for sampling the luminance of an image is 144 lines of 176 pixels, this corresponding to a format of 11 times 9 macro-blocks. For the chrominance and for each of the colour components, the format is 72 lines of 88 pixels. This corresponds to a so-called 4:1:1 luminance chrominance structure.

This image format is also called the CIQF format (standing for Common Intermediate Quarter Format).

In this example it will be assumed that such an image format is available at the input 1 of the coder.

Hereafter, for the purposes of simplifying the explanations, only the luminance will be considered.

Figure 2 gives the numbering of the luminance macro-blocks. The actual macro-blocks of an image are surrounded by

- 10 -

continuous lines. Hereafter, the eight macro-blocks which surround a current macro-block (that is to say the one currently being processed) will be referred to as adjacent macro-blocks. The macro-blocks adjacent to the macro-block 13 are the macro-blocks 1, 2, 3, 12, 14, 23, 24 and 25. Again for the purposes of simplifying the explanations, the presence of dummy macro-blocks beyond the boundary of the image will be assumed. Macro-block A is a macro-block adjacent to the macro-blocks 1, 12 and 23.

For each macro-block coded in "inter" mode, the processor 10 provides a pair of components of the motion vector associated with this block. In accordance with the present embodiment, the components have a precision of half a pixel and are confined to the interval [-16, +15.5].

According to the present embodiment, the circuit 14, which comprises for example a memory and a suitably programmed microprocessor, determines a prediction vector for each motion vector. The components Px and Py of this prediction vector are subtracted from the components of the motion vector to be coded.

According to the present embodiment, the components of the prediction vector are calculated in a manner made explicit by Figure 3.

M_0 denotes the current macro-block, that is to say the macro-block for which a prediction vector is being determined. On the other hand, MV_{ix} and MV_{iy} denote the first and second components respectively of the motion vector associated with macro-block i .

M_1 denotes the macro-block immediately to the left of M_0 ,

- 11 -

M2 the macro-block immediately above M0, and M3 the macro-block above and to the left of M0. The motion vectors associated with the macro-blocks M1 to M3 are called candidate vectors and are denoted MV1 to MV3. Figure 3a gives the arrangement of the macro-blocks M1 to M3 around M0.

Since a motion vector is defined only for macro-blocks coded in "inter" mode, the components of the candidate vectors are assumed to be zero in the following cases:

- when a macro-block has been coded in "intra" mode,
- when a macro-block lies beyond the boundary of the image.

Figures 3b, 3c and 3d illustrate various cases. The pairs (0,0) indicate the macro-blocks for which the candidate vector is regarded as zero. The dotted line indicates the boundaries of the image.

Having determined the components of the three candidate vectors, the components of the prediction vector are calculated by choosing for each component of this vector the median value of the corresponding components of the candidate vectors. For example, if the candidate vectors are respectively:

$$MV1 = (-2; 3), MV2 = (1; 5), MV3 = (-1; 7)$$

then the prediction vector will be $(Px; Py) = (-1; 5)$.

We then form the difference between this prediction vector and the motion vector M0:

$$MVDx = MVx - Px$$

$$MVDy = MVy - Py.$$

These differences are next coded by VLC coding and

2160087

- 12 -

introduced into the video multiplex.

Given that the calculation of the prediction vector demands a knowledge of the vectors M1 to M3, these vectors are held in memory by the processor 10.

During decoding, a prediction vector is determined in a similar manner. Of course, addition of the components of this vector with the decoded differences will be carried out.

Figure 4 gives a functional diagram of a decoder. The data received from a reception and demodulation system are stored in a register 15. This register is required in order to match the speed of transmission of the data to their actual use. The data are demultiplexed by a demultiplexer 16. The coefficients are decoded, undergo inverse quantization (circuit 17), and then an inverse DCT transformation (circuit 18). The resulting blocks are added (by the adder 19) to the result of the prediction in the case of coding in "inter" mode. The prediction memory 21 is of course updated appropriately.

A processor 20 is tasked with controlling the quantization and in decoding the data corresponding to the motion vectors. It carries out the VLC decoding of the data, retains in memory the vectors required for the calculation of the prediction vector, determines this vector and then adds to the components thereof the decoded differences of components. The motion vectors thus re-established are used in a known manner.

According to an embodiment variant, motion vectors are associated with the macro-blocks coded in "intra" mode. This makes it possible to compensate for the transmission errors which could

- 13 -

corrupt the coded coefficients of this macro-block. In this case, the corresponding candidate vectors are not regarded as being zero.

Obviously, the present invention is not limited to the embodiment example described, but applies to other compression systems using motion compensation.

2160087

- 14 -

Claims

1. Process for the differential coding of a motion vector (MV0) associated with a macro-block (M0), characterized in that it comprises the steps:

- of determining the components (MV0x, MV0y) of the said motion vector (MV0),
- of determining a prediction vector, each component (Px, Py) of which is equal to the median value of the corresponding components of at least three candidate motion vectors (MV1, MV2, MV3), a candidate motion vector being a vector associated with a previously coded macro-block,
- of subtracting the components thus obtained from the components of the vector to be coded.

2. Process according to Claim 1, characterized in that the components of a candidate motion vector are regarded as being zero when determining the prediction vector if the macro-block associated with this candidate motion vector lies outside the image.

3. Process according to either of Claims 1 and 2, characterized in that the components of a candidate motion vector are regarded as being zero when determining the prediction vector if the macro-block associated with this candidate motion vector was coded in "intra" mode.

4. Process according to one of the preceding claims, characterized in that the macro-blocks associated with the candidate motion vectors are macro-blocks adjacent to the macro-block of which a vector is to be coded.

- 15 -

5. Process according to one of the preceding claims, characterized in that the macro-blocks associated with the candidate motion vectors form part of the macro-blocks lying above or to the left of the macro-block of which a vector is to be coded.

6. Process according to claim 5, characterized in that the macro-blocks associated with the candidate motion vectors comprise the macro-block lying immediately to the left, the macro-block lying immediately above and the macro-block lying above and to the right of the macro-block of which a vector is to be coded.

7. Process for the decoding of a motion vector (MV0) coded by differential coding, characterized in that it comprises the steps:

- of extracting the differences in components (MVDx, MVDy) corresponding to this vector (MV0),

- of determining a prediction vector, each component (Px, Py) of which is equal to the median value of the corresponding components of vectors associated with at least three previously decoded macro-blocks (MV1, MV2, MV3),

- of adding the said differences in components (MVDx, MVDy) to the components (Px, Py) of the prediction vector.

8. Process according to Claim 7, characterized in that the components of a candidate motion vector are regarded as being zero when determining the prediction vector if the macro-block associated with this candidate motion vector lies outside the image.

9. Process according to either of Claims 7 and 8, characterized in that the components of a candidate motion vector

- 16 -

are regarded as being zero when determining the prediction vector if the macro-block associated with this candidate motion vector was coded in "intra" mode.

10. Process according to one of Claims 7 to 9, characterized in that the macro-blocks associated with the candidate motion vectors are macro-blocks adjacent to the macro-block of which a vector is to be decoded.

11. Process according to one of Claims 7 to 10, characterized in that the macro-blocks associated with the candidate motion vectors form part of the macro-blocks lying above or to the left of the macro-block of which a vector is to be decoded.

12. Process according to claim 11, characterized in that the macro-blocks associated with the candidate motion vectors comprise the macro-block lying immediately to the left, the macro-block lying immediately above and the macro-block lying above and to the right of the macro-block of which a vector is to be decoded.

13. Image compression device including a motion estimation processor (10) producing motion vectors (MV0), characterized in that, with the said vectors being coded with respect to a prediction vector, the said device comprises:

- means (10) for determining a prediction vector, each component (Px, Py) of which is equal to the median value of the corresponding components of at least three candidate motion vectors (MV1, MV2, MV3), a candidate motion vector being a vector associated with a previously coded macro-block,

- means (10) for subtracting the components thus obtained from the components of the vector to be coded,

- 17 -

- means (14) for coding the values (MVDx, MVDy) obtained through the said subtraction.

14. Device according to Claim 13, characterized in that it implements the process according to one of Claims 1 to 6.

15. Device for decompressing images compressed with the aid of a motion compensation process, characterized in that it comprises:

- a buffer register (15) receiving compressed data,
- demultiplexing means (16) connected to the said buffer (15),

- dequantization means (17) connected to the demultiplexing means (16),

- means (18) of inverse transformation of demultiplexed and dequantized coefficients,

- means (19) for extracting data relating to the motion vectors coded according to a differential coding with respect to a prediction vector,

- means (20) for determining the prediction vector on the basis of the motion vectors corresponding to previously decoded macro-blocks.

16. Device according to Claim 15, characterized in that it implements the process according to one of Claims 7 to 12.

Fetherstonhaugh & Co.,
Ottawa, Canada
Patent Agents

2160087

1/4

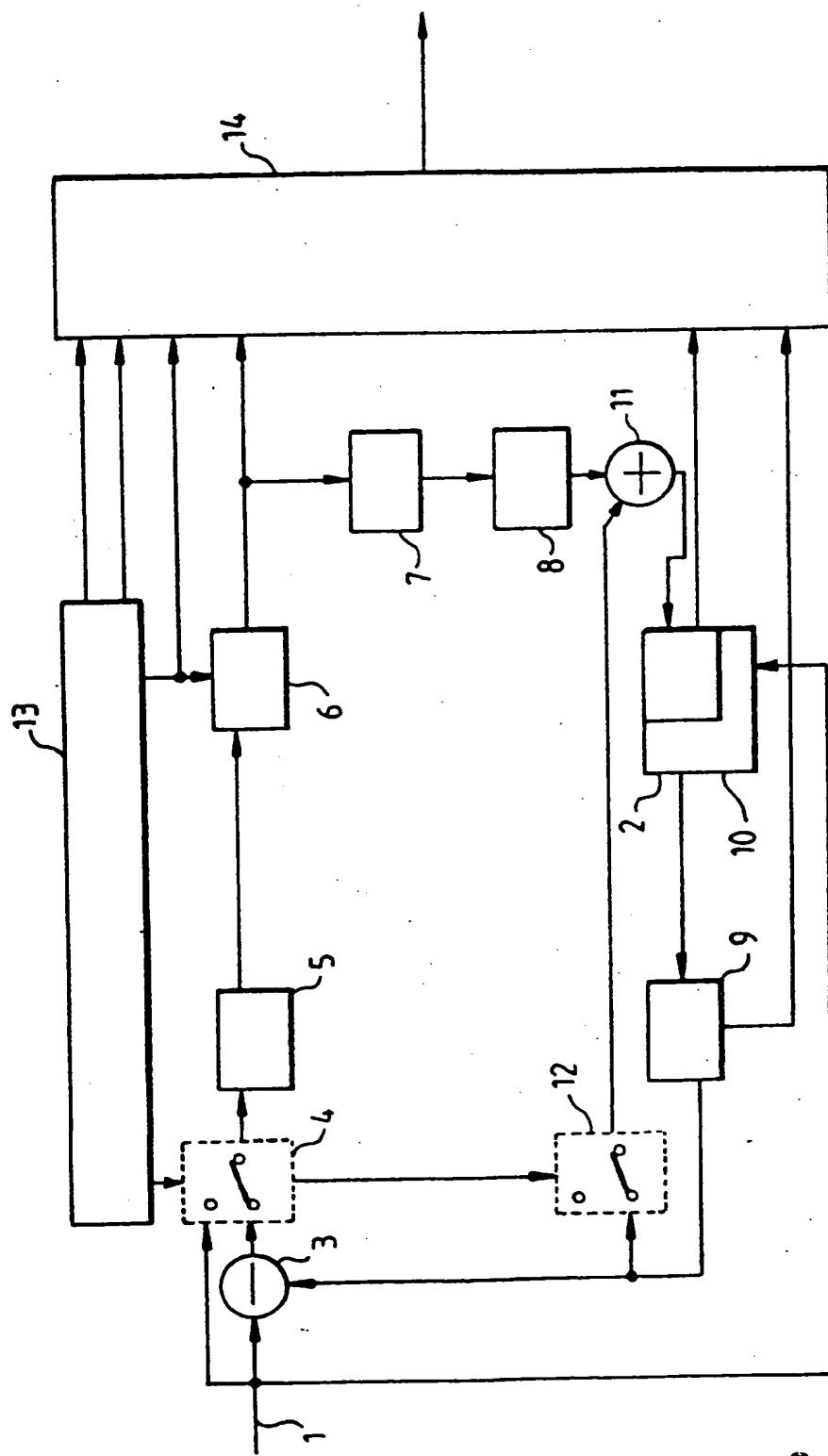


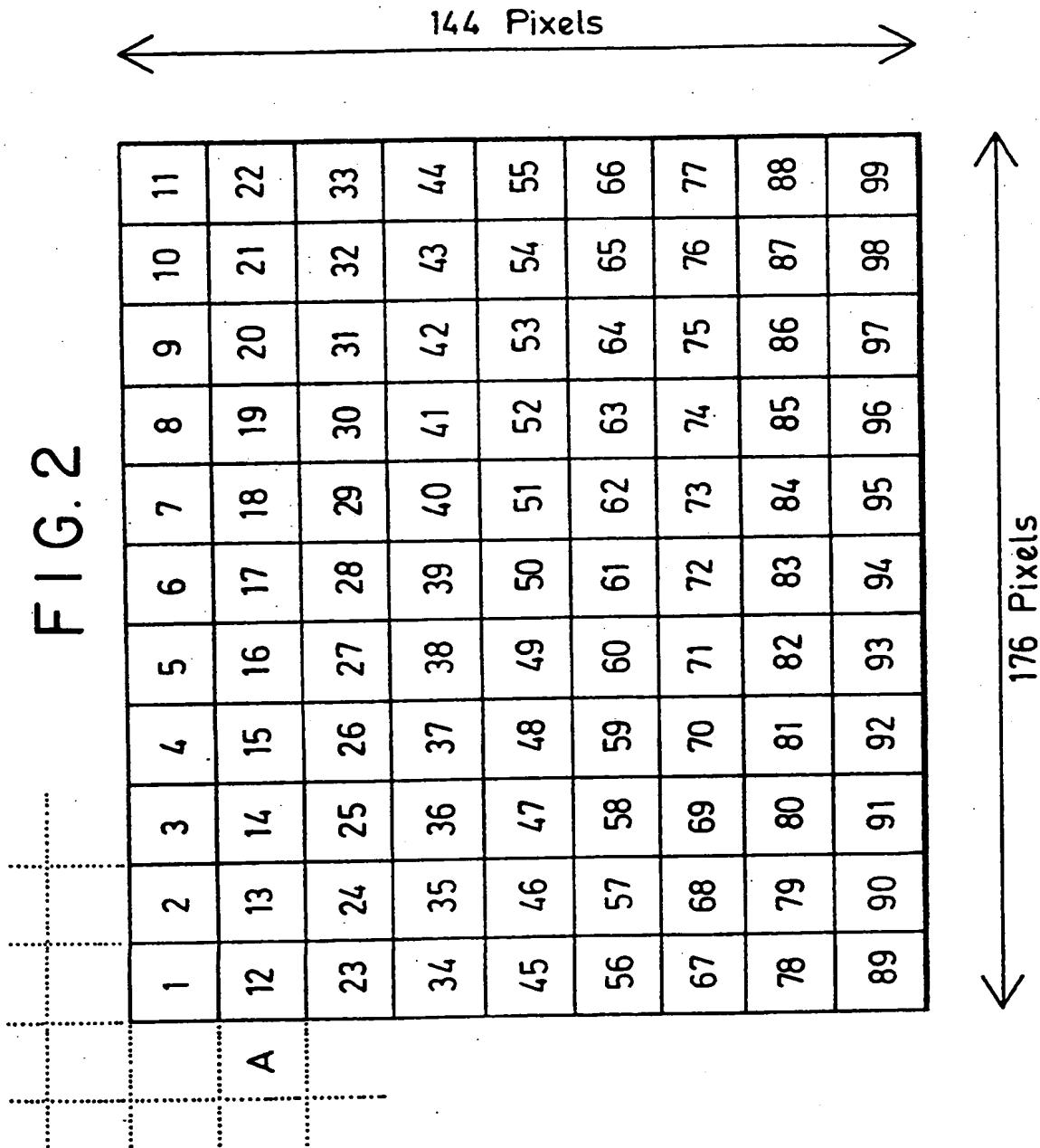
FIG. 1

Patent Agents
Fotheringham & Co.

2160087

2/4

FIG. 2



Patent Appl.

100-1000

2160087

3/4

3a

	M 2	M 3
M 1	M 0	

	MV2	MV3
(0,0)	MV0	

3b

	(0,0)	(0,0)
MV1	MV0	

3c

	MV2	(0,0)
MV1	MV0	

3d

FIG. 3

Patent No. 2160087
Fetzerste, Inc., 1980

2160087

414

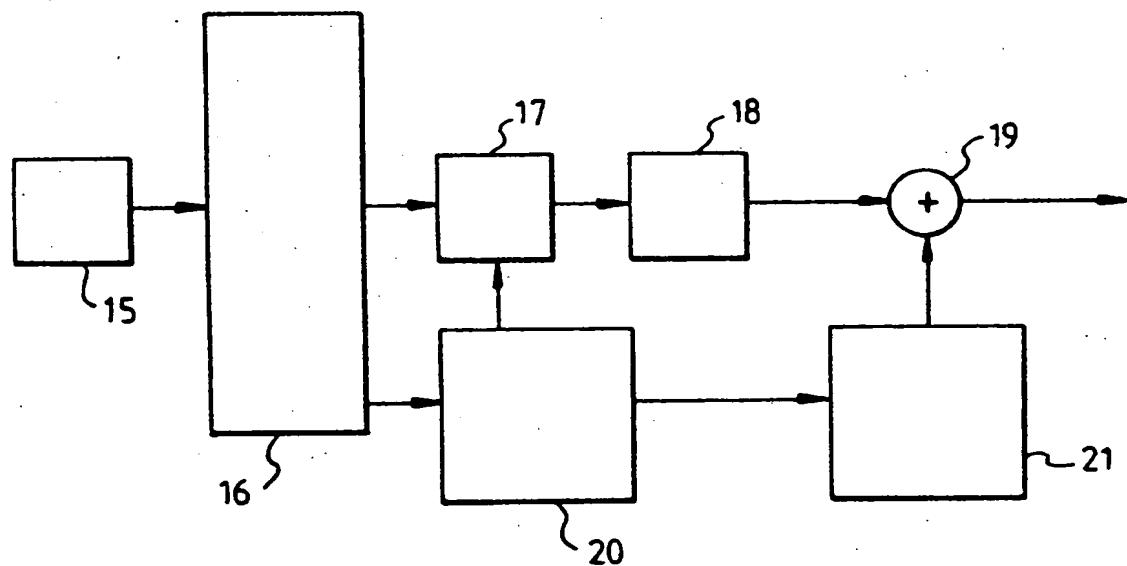


FIG. 4

Patent Agents
Fetherstonhaugh & Co.

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- BLACK BORDERS**
- IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- FADED TEXT OR DRAWING**
- BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- SKEWED/SLANTED IMAGES**
- COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- GRAY SCALE DOCUMENTS**
- LINES OR MARKS ON ORIGINAL DOCUMENT**
- REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.